

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: JOHN EZELL ET AL.)
Serial No.: 10/696,914) Group Art Unit: 2163
Filed: October 29, 2003)
For: SYSTEM AND METHOD FOR) Examiner:
SYNCHRONIZING DATA IN A NETWORKED) Darno
SYSTEM)

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

REAL PARTY IN INTEREST

The real party in interest is AT&T Delaware Intellectual Property, Incorporated, formally known as BellSouth Intellectual Property Corporation, the assignee of record.

RELATED APPEALS AND INTERFERENCES

There are no pending appeals or interferences related to this appeal.

STATUS OF CLAIMS

Claims 1-4, 6-10 and 12 were finally rejected under 35 U.S.C. § 102(b) as being anticipated by Beeler.

Claims 5 and 11 were finally rejected under 35 U.S.C. § 103 as being unpatentable over Beeler in view of Ellard.

The rejections of claims 1-12 are herein appealed.

STATUS OF AMENDMENTS

There have been no amendments filed after the final rejection mailed August 22, 2007.

SUMMARY OF CLAIMED SUBJECT MATTER

A concise explanation of the subject matter defined in each of the independent claims involved in the appeal is provided below.

Embodiments relate to synchronizing data sets between first and second computer servers. The data sets include an identifier and a set of attributes as shown, for example, in Applicants' Figure 2. In order to identify data sets that are modified, and the need for synchronization, when an attribute of a data set is modified a checksum associated with that data set is set to a predetermined value (e.g., zero) as described in paragraph [0024]. The next step involves accessing data sets having the predetermined checksum and formatting an attribute for that data set to a predetermined format type (paragraph [0025]), which is shown in Figure 2 as an intermediate format type. To synchronize first data sets and second data sets, the checksums are compared. If the checksums do not match, the data sets from the first computer server are transmitted to the second computer server to replace the second data sets.

Independent claim 1 recites a method for synchronizing data in first and second computer servers, the first computer server (Figure 1, element 12) including a first plurality of data sets (Figure 2, elements 42 and 44; paragraph [0017]) each having a first identifier (Figure 2; paragraph [0019]) and a first set of attributes (Figure 2; paragraph [0019]), and the second computer server (Figure 1, element 22) including a second plurality of data sets (Figure 2, element 46; paragraph [0017]) each having a second identifier (Figure 2; paragraph [0020]) and a second set of attributes (Figure 2; paragraph [0020]), the method comprising: modifying an attribute of one of the first plurality of data sets (Figure 3B, element 64; paragraph [0023]) and setting a checksum associated with one of the first plurality of data sets to a predetermined value (Figure 3B, element 66; paragraph [0024]); accessing one of the first plurality of data sets having a checksum set to the predetermined value and formatting at least one attribute associated with the one of the first plurality of data sets to a 030366 (BLL-0119)

predetermined format type (Figure 3B, element 68; paragraph [0025]); transmitting a second identifier and a second checksum value both associated with one of the second plurality of data sets to the first computer server (Figure 3C, element 72; paragraph [0028]); accessing one of the first plurality of data sets having a first identifier corresponding to the transmitted second identifier to determine a first checksum value associated with the accessed data set (Figure 3C, element 74; paragraph [0029]); and, when the first checksum value is not equal to the transmitted second checksum value (Figure 3C, element 76; paragraph [0030]), transmitting the one of the first plurality of data sets from the first computer server to the second computer server (Figure 3C, element 78; paragraph [0031]) to replace a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the one of the first plurality of data sets (Figure 3C, element 80; paragraph [0032]).

Independent claim 7 recites a networked system, comprising: a first computer server (Figure 1, element 12) and a second computer server (Figure 1, element 22) operably communicating with one another, the first computer server configured to store a first plurality of data sets (Figure 2, elements 42 and 44; paragraph [0017]) each having a first identifier (Figure 2; paragraph [0019]) and a first set of attributes (Figure 2, paragraph [0019]), and the second computer server configured to store a second plurality of data sets (Figure 2, element 46; paragraph [0017]) each having a second identifier (Figure 2; paragraph [0020]) and a second set of attributes (Figure 2; paragraph [0020]); the first computer server further configured to modify an attribute of one of the first plurality of data sets (Figure 3B, element 64; paragraph [0023]) and set a checksum associated with one of the first plurality of data sets to a predetermined value (Figure 3B, element 66; paragraph [0024]); the first computer server further configured to access one of the first plurality of data sets having a checksum set to the predetermined value and format at least one attribute associated with the one of the first plurality of data sets to a predetermined format type (Figure 3B, element 68; paragraph [0025]); the second computer server further configured to transmit a second identifier and a second checksum value both associated with one of the second plurality of data sets to the first computer server (Figure 3C, element 72; paragraph [0028]); the first computer server further configured to access one of the first plurality of data sets having a first identifier that corresponds to the transmitted second identifier to determine a first checksum value

associated with the accessed data set (Figure 3C, element 74; paragraph [0029]); and, the first computer server further configured to transmit the one of the first plurality of data sets from the first computer server to the second computer server (Figure 3C, element 78; paragraph [0031]) to replace a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the one of the first plurality of data sets (Figure 3C, element 80; paragraph [0032]) when the first checksum value is not equal to the transmitted second checksum value (Figure 3C, element 76; paragraph [0030]).

The above exemplary embodiments are discussed with respect to the aforementioned independent claims by way of example only and are not intended to in any way limit the scope of these claims.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-4, 6-10 and 12 finally rejected under 35 U.S.C. § 102(b) as being anticipated by Beeler.

Claims 5 and 11 finally rejected under 35 U.S.C. § 103 as being unpatentable over Beeler in view of Ellard.

ARGUMENT

1. Claims 1-4, 6-10 and 12 are novel over Beeler

Claims 1-4, 6-10 and 12 were rejected under 35 U.S.C. § 102(b) as being anticipated by Beeler. This rejection is traversed for the following reasons.

Claim 1 as amended recites, *inter alia*, “modifying an attribute of one of the first plurality of data sets and setting a checksum associated with one of the first plurality of data sets to a predetermined value; accessing one of the first plurality of data sets having a checksum set to the predetermined value and formatting at least one attribute associated with the one of the first plurality of data sets to a predetermined format type.”

Embodiments of the invention detect when an attribute is modified and identify that data element by setting a checksum to a predetermined value such as zero (see paragraph [0024]). The data sets having the checksum of zero are then converted to 030366 (BLL-0119)

predetermined format types (paragraph [0025]). This allows the system to efficiently sweep through data and find items needing synchronization. Beeler fails to teach these features.

Beeler discloses a fast-mirroring process between a source server and a target server. The fast-mirroring process breaks files into blocks and computes a checksum for each block. The checksums are used to determine if the files on the source server match the target server (see Figures 26 and 27, column 15, line 46 – column 16, line 7). There is no teaching in Beeler of “modifying an attribute of one of the first plurality of data sets and setting a checksum associated with one of the first plurality of data sets to a predetermined value; accessing one of the first plurality of data sets having a checksum set to the predetermined value and formatting at least attribute associated with the one of the first plurality of data sets to a predetermined format type” as recited in claim 1. There is no setting a checksum to a predetermined value in Beeler. Beeler computes checksums for blocks of data to determine if they are equal and need synchronizing. There is no teaching in Beeler of setting a checksum to a predetermined value when an attribute of a data set is modified as recited in claim 1. The checksums in Beeler are computed for the blocks of data and there is no predetermined value used each time a block of data is changed. In the Examiner’s response to Applicants’ remarks, the Examiner suggests that Beeler does teach a predetermined checksum when data sets are modified, and cites to column 15, lines 53-56 which only discusses computing a checksum for a block of data. If the checksum in Beeler must be computed, then there is no teaching in Beeler of using a predetermined value for a checksum when a data set is modified.

Further, claim 1 recites, “accessing one of the first plurality of data sets having a checksum set to the predetermined value and formatting at least attribute associated with the one of the first plurality of data sets to a predetermined format type.” In applying Beeler, the Examiner cites to two sections of Beeler that generally recite a “format” for data. There is no mention that “data sets having a checksum set to the predetermined value” are formatted to a predetermined format type, nor is there any relation taught by Beeler between the checksum value and the format of the data. Beeler simply fails to teach this element of claim 1. The Examiner has essentially found the word “format” in

sections of Beeler and relied on this word to allegedly anticipate this element of claim 1. Beeler fails to teach all the elements of claim 1 and thus cannot anticipate claim 1.

For at least the above reasons, claim 1 is patentable over Beeler. Claims 2-4 and 6 variously depend from claim 1 and are patentable over Beeler for at least the reasons advanced with reference to claim 1.

Independent claim 7 recites features similar to those discussed above with reference to claim 1 and is patentable over Beeler for at least the reasons advanced with reference to claim 1. Claims 8-10 and 12 depend from claim 7 and are considered patentable for at least the same reasons.

II. Claims 5 and 11 are non-obvious and are patentable

Claims 5 and 11 were rejected under 35 U.S.C. § 103 as being unpatentable over Beeler in view of Ellard. This rejection is traversed for the following reasons. Ellard was relied upon for disclosing converting data from a first format to a second format, but fails to cure the deficiencies of Beeler discussed above with reference to claim 1. Ellard only discloses a data exchange that converts two types of data. Ellard is not directed to synchronizing first and second servers and does not teach use of checksums as discussed above with reference to claim 1. Claim 5 depends from claim 1 and is patentable over Beeler in view of Ellard for at least the reasons advance with reference to claim 1. Claim 11 depends from claim 7 and is patentable over Beeler in view of Ellard for at least the reasons advance with reference to claim 7.

II. Conclusion

In view of the foregoing, it is respectfully submitted that the application is in condition for allowance. Accordingly, it is respectfully requested that this application be allowed and a Notice of Allowance issued.

In the event the Commissioner of Patents and Trademarks deems additional fees to be due in connection with this application, Applicants' attorney hereby authorizes that such fee be charged to Deposit Account No. 06-1130.

Respectfully submitted,

By:

David A. Fox
Registration No. 38,807
CANTOR COLBURN LLP
20 Church Street, 22nd Floor
Hartford, CT 06103-3207
Telephone (860) 286-2929
Facsimile (860) 286-0115
Customer No. 36192

Date: June 16, 2008

CLAIM APPENDIX

1. A method for synchronizing data in first and second computer servers, the first computer server including a first plurality of data sets each having a first identifier and a first set of attributes, and the second computer server including a second plurality of data sets each having a second identifier and a second set of attributes, the method comprising:

modifying an attribute of one of the first plurality of data sets and setting a checksum associated with one of the first plurality of data sets to a predetermined value;

accessing one of the first plurality of data sets having a checksum set to the predetermined value and formatting at least one attribute associated with the one of the first plurality of data sets to a predetermined format type;

transmitting a second identifier and a second checksum value both associated with one of the second plurality of data sets to the first computer server;

accessing one of the first plurality of data sets having a first identifier corresponding to the transmitted second identifier to determine a first checksum value associated with the accessed data set; and,

when the first checksum value is not equal to the transmitted second checksum value, transmitting the one of the first plurality of data sets from the first computer server to the second computer server to replace a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the one of the first plurality of data sets.

2. The method of claim 1 wherein the one of the first plurality of data sets includes the first set of attributes associated with one of a physical network element, a software event, and a logical operator.

3. The method of claim 1 wherein the first checksum value is calculated by the first computer server using the first set of attributes associated with the one of the first plurality of data sets.

4. The method of claim 1 further comprising transmitting the first checksum value to the second computer server to replace the second checksum value in the second computer server with the first checksum value.

5. The method of claim 1 wherein the one of the first plurality of data sets is formatted differently than the one of the second plurality of data sets.

6. The method of claim 1 wherein the first plurality of data sets includes at least one data set that does not correspond to any of the second plurality of data sets.

7. A networked system, comprising:

a first computer server and a second computer server operably communicating with one another, the first computer server configured to store a first plurality of data sets each having a first identifier and a first set of attributes, and the second computer server configured to store a second plurality of data sets each having a second identifier and a second set of attributes;

the first computer server further configured to modify an attribute of one of the first plurality of data sets and set a checksum associated with one of the first plurality of data sets to a predetermined value;

the first computer server further configured to access one of the first plurality of data sets having a checksum set to the predetermined value and format at least one attribute associated with the one of the first plurality of data sets to a predetermined format type;

the second computer server further configured to transmit a second identifier and a second checksum value both associated with one of the second plurality of data sets to the first computer server;

the first computer server further configured to access one of the first plurality of data sets having a first identifier that corresponds to the transmitted second identifier to determine a first checksum value associated with the accessed data set; and,

the first computer server further configured to transmit the one of the first plurality of data sets from the first computer server to the second computer server to replace a second set of attributes of the one of the second plurality of data sets with the first set of attributes of the one of the first plurality of data sets when the first checksum value is not equal to the transmitted second checksum value.

8. The networked system of claim 7 wherein the one of the first plurality of data sets includes a first set of attributes associated with one of a physical network element, a software event, and a logical operator.

9. The networked system of claim 7 wherein the first checksum value is calculated by the first computer server using the a first set of attributes associated with the one of the first plurality of data sets.

10. The networked system of claim 7 wherein the first computer is further configured to transmit the first checksum value to the second computer server to replace the second checksum value in the second computer server with the first checksum value.

11. The networked system of claim 7 wherein the one of the first plurality of data sets is formatted differently than the one of the second plurality of data sets.

12. The networked system of claim 7 wherein the first plurality of data sets includes at least one data set that does not correspond to any of the second plurality of data sets.

EVIDENCE APPENDIX

Not Applicable

RELATED PROCEEDINGS APPENDIX

Not Applicable